

[oracle.com](https://www.oracle.com)

## GlassFish and MySQL, Part 1: A Perfect Combination for Web Applications

33-41 minutes

---

With more than 100 million downloads, [MySQL](#) is the world's most popular open-source database. MySQL's popularity is indicative of the growing adoption of open-source software. Developers are using open-source software because it offers them a reliable and low-cost alternative for developing their applications. This adoption trend extends to middleware too. For example, open-source servers are replacing proprietary servers in many enterprises. Case in point: [GlassFish](#), an open-source, enterprise-quality, Java EE 5-compliant application server, enjoys significant popularity. With more than seven million downloads since its release in May 2005 and more than half a million downloads a month, GlassFish has a widespread and growing community of users.

The reasons that so many people download and use MySQL and GlassFish are compelling. In addition to being open source, MySQL and GlassFish are fast, reliable, and easy to use. Though attractive individually, MySQL and GlassFish when used together provide an excellent solution for quickly developing and deploying web applications that are not only secure and reliable, but also scale to meet increasing demand and are highly responsive to user requests.

When used together, MySQL and GlassFish provide an excellent, low-cost solution for quickly developing and deploying web applications that are not only secure and reliable, but also scale to meet increasing demand and are highly responsive to user requests.

This article, the first in a series of articles, covers the advantages of

using MySQL with GlassFish and illustrates why the combination is a perfect choice for developing and deploying web applications. In future articles in the series, you'll see how easy it is to build web applications that take advantage of MySQL and GlassFish features.

Although various versions of GlassFish and MySQL are available for download, this series of articles focuses on GlassFish v2 Update Release 2 (UR2) and [MySQL 5.1 Community Server](#).

## **What Makes MySQL and GlassFish Popular**

Many of the characteristics that make MySQL so successful also apply to GlassFish. This section briefly describes those characteristics.

### **Open Source**

One of the characteristics that immediately attracts people to MySQL is that it is open-source software. This type of software is backed by an open-source license that gives users -- typically, program developers -- the right to access source code and change it to meet their needs. In addition, open-source software stimulates a collaborative environment where users participate in sharing their code, upgrades, and ideas.

The open-source license that applies to MySQL is the GNU General Public License (GPL). Through GPL, users can view the MySQL source code, change it, and share their changes with an active community of open source developers and enthusiasts. By participating in the MySQL open source community, users can get the latest community-based upgrades, bug fixes, and innovations.

MySQL Community Server and GlassFish v2UR2 are open source. Both are backed by a large and active community of users who provide community-based upgrades, bug fixes, and support.

As is the case for MySQL, GlassFish is open-source software. GlassFish source code is available under dual licenses: the Common Development and Distribution License (CDDL) and GPL Version 2 (GPL v2) plus Classpath exception. Users can choose

the license that best fits their needs. Either of these open-source licenses allows users to access the GlassFish source code and customize it to meet their requirements.

GlassFish too is supported by a large and active community. Many members of the community work on the GlassFish project, whose primary objective is developing the open-source GlassFish application server. In addition, the community participates in GlassFish subprojects, where they develop various technologies apart from the application server. Members of the GlassFish community don't necessarily develop code for the application server. By simply using the application server, they can file and track bugs and exchange information with other community members. Members of the GlassFish community have access to the latest community-based upgrades, bug fixes, and support.

### Easy to Download and Install

[link](#) You can download and install MySQL quickly -- usually in minutes. The process typically involves downloading an appropriate [MySQL server distribution package](#), running the setup program in the downloaded distribution, and performing any necessary post installation setup such as creating grant tables for user account information. Some postinstallation actions are unnecessary, depending on the environment. For example, you don't have to create the grant tables for Microsoft Windows environments because the MySQL distributions for Windows include the grant tables with a set of preinitialized accounts.

As fast and easy as it is to install MySQL, it might be even faster and easier to download and install GlassFish. All you need to do is download the appropriate GlassFish distribution package, run the installer in the downloaded package, and then run a prepackaged setup script.

Although you can download and install MySQL and GlassFish individually, you can also download them in a single [bundle that includes MySQL 5.0 Community Server and GlassFish v2ur1](#). After you download the bundle, you run a setup script to install GlassFish

and MySQL Community Server. Then you configure MySQL. In this way, you can install both MySQL and GlassFish very quickly.

In addition, a [MySQL-GlassFish-NetBeans bundle](#) is available that combines MySQL 5.0 Community Server, GlassFish v2UR2, and the popular development tool, [NetBeans IDE 6.1](#). Here too the [download and installation process](#) is relatively short and easy.

## Easy to Use

MySQL supports ANSI standard SQL, the standard programming language for querying and managing relational databases. MySQL supports the same language and, in general, the same feature set as other databases that support the standard. So developers have an easy learning curve as they go from their previous database system to MySQL.

MySQL supports ANSI standard SQL. Many users find SQL an easy language to use because of its relatively straightforward syntax.

Many developers find SQL an easy language to use because of its relatively straightforward syntax. In general, MySQL statements follow a "Do *action* To *target* " syntax, where *action* is the action you want to perform, and *target* is the target on which you want to perform the action. For example, to retrieve all the data in the name column from a table named `pet`, you enter the following statement:

```
select name from pet;
```

MySQL provides a client program called the MySQL monitor that allows users to enter SQL statements. To start the program you enter the `mysql` command on a command line. You can then enter SQL statements at the MySQL monitor prompt. For example, to select all the data from the `pet` table, you enter the following statement:

```
mysql> select * from pet;
```

The MySQL monitor then displays the result of the query:

Copied to Clipboard

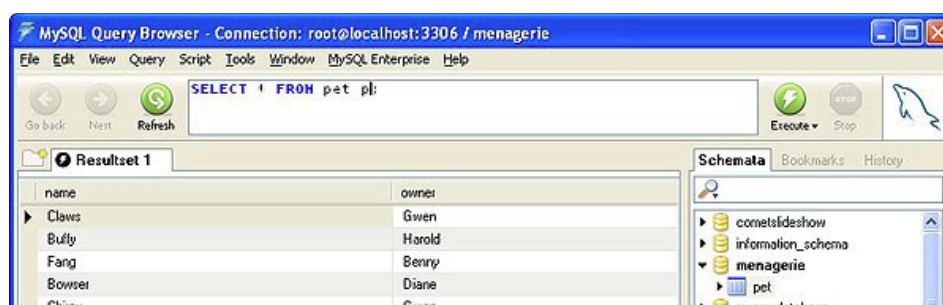
Error: Could not Copy

```

+-----+-----+-----+-----+-----+-----+
| name   | owner  | species | sex   | birth |
| death |
+-----+-----+-----+-----+-----+-----+
| Claws  | Gwen   | cat     | m     |
1994-03-17 | 0000-00-00 |
| Buffy  | Harold | dog     | f     |
1989-05-13 | 0000-00-00 |
| Fang   | Benny  | dog     | m     |
1990-08-27 | 0000-00-00 |
| Bowser | Diane  | dog     | m     |
1979-08-31 | 1995-07-29 |
| Chirpy | Gwen   | bird    | f     |
1998-09-11 | 0000-00-00 |
| Whistle | Gwen   | bird    | NULL  |
1997-12-09 | 0000-00-00 |
| Slim   | Benny  | snake   | m     |
1996-04-29 | NULL    |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.02 sec)

```

MySQL also offers a tool called the MySQL Query Browser that allows you to graphically build and manage queries. For example, [Figure 1](#) shows the MySQL Query Browser user interface (UI). If a user drags the `pet` table from the object browser to the query area, the MySQL Query Browser automatically creates a query in the query area that retrieves all the data from the `pet` table. Double clicking the table in the object browser also generates the query. If a user clicks the Execute button, the MySQL Query Browser executes the query and displays the result, as shown in Figure 1.



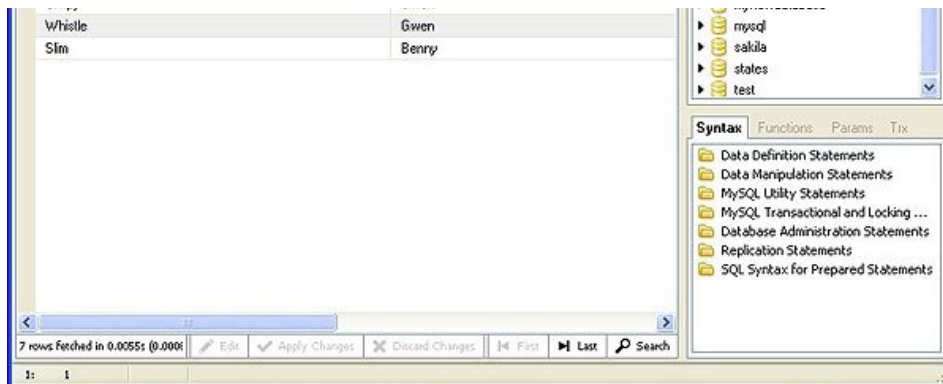


Figure 1. Building and Executing a Query in the MySQL Query Browser

Ease of use is also an important objective of GlassFish. In fact, ease of use is the primary objective of [Java EE 5](#), the Java platform version that GlassFish fully implements. Because it is open source, GlassFish gives developers the flexibility to meet their needs. And by fully supporting the Java EE 5 standard, GlassFish provides the technical richness that's required to provide real-life solutions. It's this combination of characteristics that give developers true freedom.

Because it is open source, GlassFish gives developers the flexibility to meet their needs. And by fully supporting the Java EE 5 standard, GlassFish provides the technical richness that's required to provide real-life solutions. It's this combination of characteristics that give developers true freedom.

An important usability enhancement in the Java EE 5 platform is the use of annotations to mark objects -- or methods within objects -- as having particular characteristics or to inject dependencies on resources. Various Java EE 5 technologies implemented in GlassFish such as [Java Servlet 2.5](#), [JavaServer Faces \(often referred to as JSF\) 1.2](#), [Enterprise JavaBeans \(EJB\) 3.0](#) -- which includes the [Java Persistence API](#) -- and web services technologies such as [Java API for XML-Based Web Services \(JAX-WS\) 2.0](#) and [Java Architecture for XML Binding \(JAXB\) 2.0](#), support the use of annotations. With annotations you can simplify the code for objects such as servlet classes, EJB bean classes, and web services implementation classes, enabling you to code these objects as Plain Old Java Objects (POJOs).

GlassFish takes advantage of usability enhancements in Java EE 5 such as annotations to simplify the coding necessary to build a web application.

Resource-injection annotations replace a lot of the resource-related interactions that an object has to have with its container in the application server. For example, in the following code snippet, the `@Resource` annotation injects a data source into a servlet. The annotation frees you from having to include code to obtain a Java Naming and Directory Interface (JNDI) context from the servlet container and do a JNDI lookup for the data source.

Copied to Clipboard

Error: Could not Copy

```
public class MyServlet extends HttpServlet {  
    @Resource(name="myDs") DataSource  
    myDataSource;  
    ...  
}
```

Other annotations eliminate a lot of the boilerplate code, such as XML deployment descriptors, that previously needed to be developed for objects such as EJB entity classes and web services implementation classes. For example, the following code snippet marks an EJB bean as a remote stateless session bean:

Copied to Clipboard

Error: Could not Copy

```
@Stateless  
@Remote  
public class ConverterBean {  
    ...  
}
```

The `@Stateless` and `@Remote` annotations replace the need for you to create a home interface, remote interface, and deployment descriptor for the bean, saving you from a lot of tedious coding.

### Easy to Administer



SQL is not only an easy-to-use query language, it's also an easy-to-use data management language. You can use SQL statements to perform database administrative tasks such as creating a database, granting user access to a database, creating tables, and filling tables with data. For example, to create a database named `menagerie`, you can enter the following SQL statement using the MySQL monitor prompt:

```
mysql> create database menagerie;
```

MySQL also includes a client program called `mysqladmin` that you can use to perform administrative actions such as creating and deleting databases, shutting down the MySQL server, and viewing or killing MySQL processes.

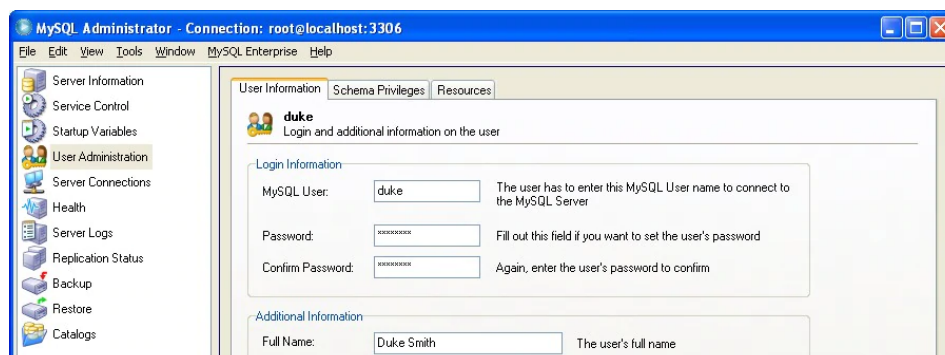
To start the program, you enter the `mysqladmin` command on a command line. Then you can enter administrative requests at the `mysqladmin` prompt. For example, the following request deletes the `menagerie` database and all its tables:

```
mysqladmin> drop database menagerie;
```

Both MySQL and GlassFish offer graphical tools that simplify administration.

MySQL also offers a graphical administrative tool called MySQL Administrator. The tool allows you to perform a variety of administrative tasks such as configuring a MySQL server and administering user accounts. It also displays critical "health" issues regarding the database server, such as the current, minimum, maximum, and average number of queries as well as the query cache hit rate.

[Figure 2](#) shows the MySQL Administrator page for viewing information about a MySQL user.





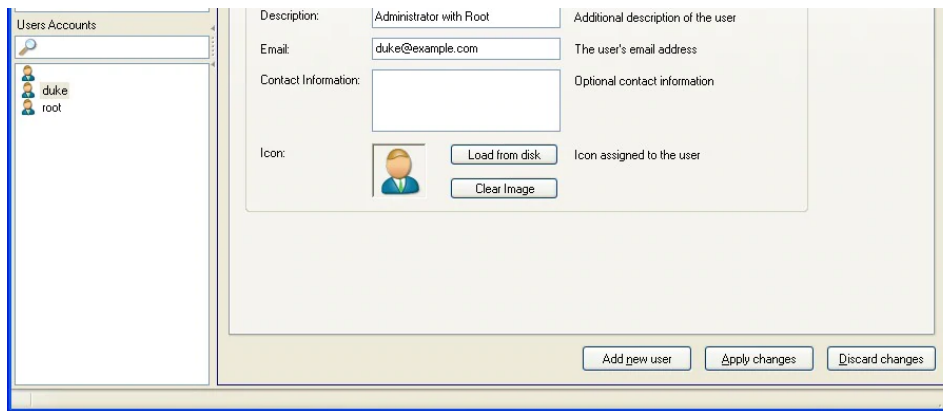


Figure 2. Viewing User Information With MySQL Administrator

GlassFish also provides easy-to-use tools for administration. One is the intelligent command line interface (CLI). The other is the Admin Console. In general, you can perform almost any GlassFish-related administrative task on a GlassFish instance or cluster through the CLI or through the Admin Console. A *cluster* is a group of GlassFish v2 application server instances that can be managed and monitored as a single logical entity.

The CLI is "intelligent" because it suggests alternative entries if you make an error when you enter a command. For example, the command to start a GlassFish domain is `asadmin start-domain`. If you mistype the keyword `domain` in the `asadmin` command as follows:

```
sadmin start-domian
```

the CLI suggests an alternative:

Copied to Clipboard

Error: Could not Copy

Closest matching command(s):

```
asadmin start-domain
```

The Admin Console provides a graphical UI to administer GlassFish instances or clusters. [Figure 3](#) shows the Common Tasks page in the Admin Console for a domain that is enabled for clustering.

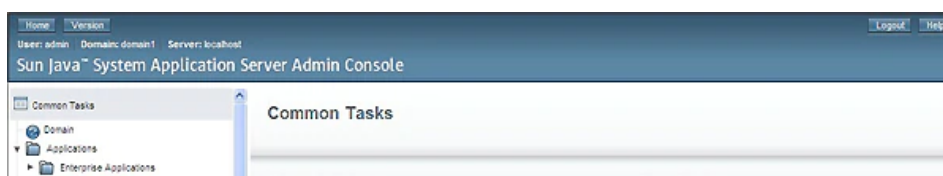




Figure 3. Admin Console Common Tasks Page

Notice that buttons are displayed for tasks such as creating a new cluster, deploying a web application, and monitoring data. These buttons are supplemented by a navigation tree that gives you an alternate path to performing administrative tasks. To initiate an operational task, you simply click the appropriate button in the page or click a node in the tree.

The Admin Console makes it particularly easy to monitor resources such as connection pools, that is, the cache of database connections maintained by the database, and also includes a call flow feature that makes it easy to track a transaction as it moves through various GlassFish application server containers. This is illustrated in [Figure 4](#).

Call Flow Steps (3)					
Calling Sequence	Container	Module	Component	Method	
1	SERVLET	helloworld	GreeterServlet	/helloworld/GreeterServlet:samples.ejb.stateless.simple.servlet.GreeterServlet.service	
2	STATELESS_SESSION_BEAN	stateless-simpleEjb.jar	TheGreeter	public abstract java.lang.String samples.ejb.stateless.simple.ejb.Greeter.getGreeting() throws java.rmi.RemoteException	
3	SERVLET	helloworld	Jsp	/helloworld/GreeterServlet:/GreeterView.jsp:org.apache.jasper.servlet.JspServlet.service	

Figure 4. Call Flow Monitoring

## Integrated With Popular Development Tools

MySQL and GlassFish are easy to integrate into developer tools such as the NetBeans IDE or the Eclipse IDE. They can also be used with popular open-source development frameworks such as Ruby on Rails, Spring, Struts, and Seam.

You can get plug-ins that enable you to integrate both MySQL and GlassFish with popular development tools such as the [NetBeans IDE](#) or the [Eclipse IDE](#). In addition, the [MySQL-GlassFish-](#)

[NetBeans bundle](#) integrates MySQL and GlassFish into NetBeans IDE 6.1. After MySQL and GlassFish are integrated into an IDE, you can use the IDE to easily register MySQL servers and perform administrative tasks such as view, create, and delete MySQL databases. You can then use the IDE to build applications that access those databases and deploy the applications on the GlassFish application server.

IDEs such as NetBeans 6.1 include features that streamline the process of building web applications, including applications that use web services or that access persistent data. For example, you can take advantage of the JavaServer Faces create, read, update, and delete (CRUD) Generator in NetBeans IDE 6.1 to create a JavaServer Faces-based CRUD application from EJB entity classes, as shown in [Figure 5](#).

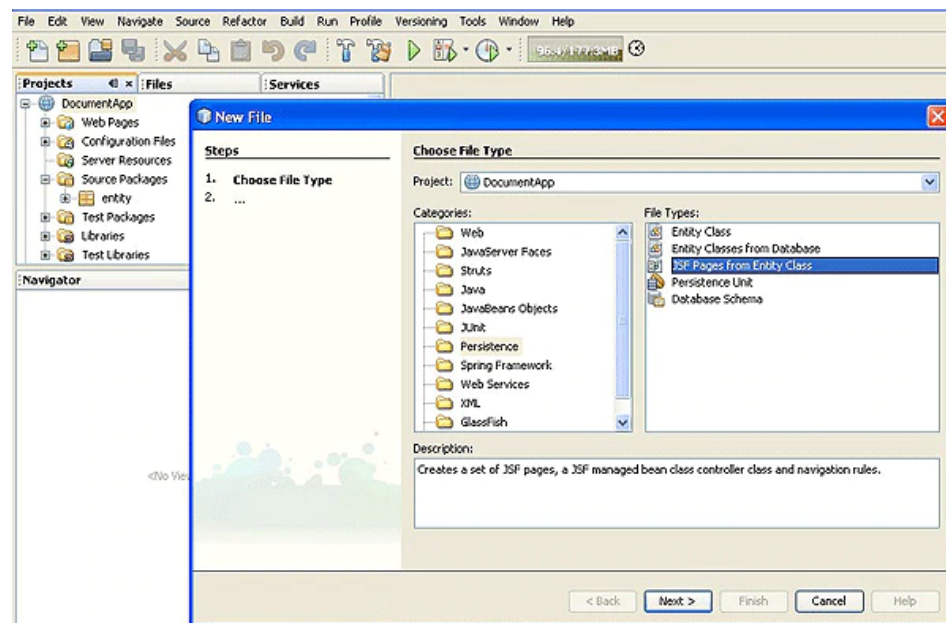


Figure 5. Creating a CRUD Application from EJB Entities in NetBeans IDE 6.1

In addition to their integration with popular IDEs, both MySQL and GlassFish can be used with popular open-source development frameworks such as [Ruby on Rails](#), [Spring](#), [Struts](#), and [Seam](#).

### Fast, Scalable, Reliable

MySQL's scale-out capabilities make it fast, scalable, and reliable.

MySQL offers scale-out capabilities that enable you to improve application performance and scalability incrementally and when you need it.

**MySQL Scale Out:** The term *scale out* refers to an architectural design that separates and spreads the database workload across different physical servers. As demand increases, you can scale out by adding database servers on low-cost commodity hardware and spread the increased workload across the database servers. This strategy is in contrast to a *scale-up* approach, which requires a large up-front investment in more expensive and complex server hardware in order to add capacity.

In a typical situation, users initiate sessions on web or application servers. These servers send requests as required into the MySQL database servers through load balancers or other application logic that directs the requests. When a request causes a server to write into the database, the server replicates its data to other MySQL servers to keep them up-to-date for subsequent queries.

[Figure 6](#) illustrates the scale-out architecture.

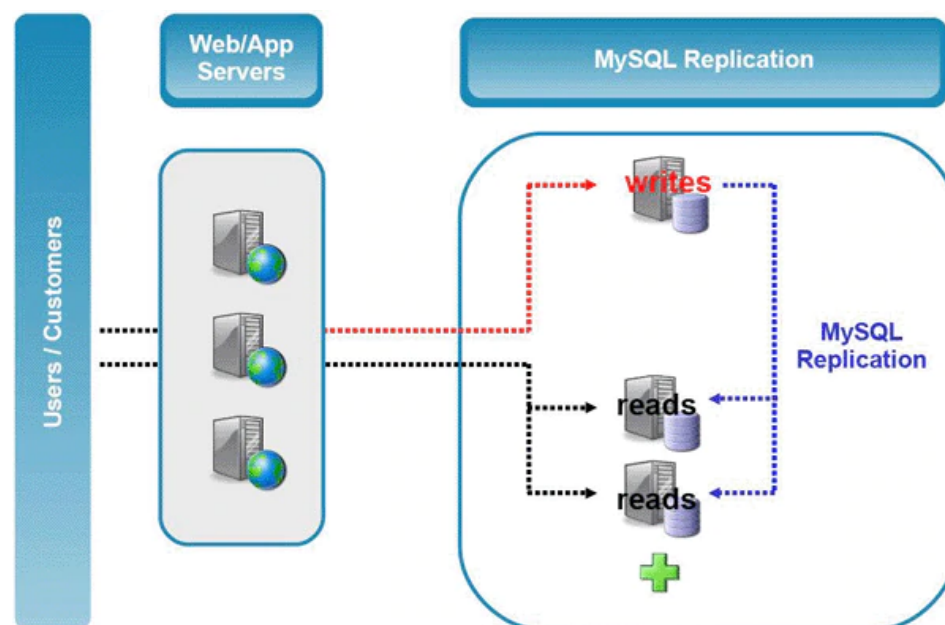


Figure 6. Scale Out Architecture in MySQL

In high-availability, high-volume settings, a failure in any MySQL server is typically corrected by taking the failed server offline and swapping it for another server. In a few minutes, the database can be replicated on the new server, and query service can proceed as

before. The low cost of commodity servers makes this swap-out strategy cost-effective.

This scale-out design helps MySQL run very fast -- the MySQL database consistently scores among the [top databases for performance and throughput](#) -- and it also makes MySQL very reliable because redundancy is built into the design.

GlassFish v2 is an extremely fast open-source application server. It takes advantage of the New I/O (NIO) APIs in the Java platform to scale to thousands of users, and ensures reliability through clustering and memory replication.

**GlassFish -- High Performance and Reliability:** GlassFish too scores consistently high for performance and throughput when compared to other application servers. In fact, recently GlassFish v2 posted the [best open source result on SPECjAppServer 2004, 1](#) -- an end-to-end application that exercises all major Java EE technologies implemented by compliant application servers -- for a pair of Sun Fire X4150 servers, each equipped with one Intel Xeon CPU. For details, see [Open Source Leader in Application and Database tiers of SPECjAppServer 2004 benchmark](#).

In general, GlassFish is the fastest open-source application server -- even outperforming most closed-source application servers, and is the industry's price-performance leader. In addition, the HTTP layer of the web tier in GlassFish, which is implemented through the Grizzly framework, takes advantage of the [Java New I/O \(NIO\) APIs](#) to scale to thousands of users.

GlassFish ensures reliability through clustering and memory replication. As mentioned earlier, a cluster is a group of GlassFish v2 application server instances that you can manage and monitor as a single logical entity. *Memory replication* is an aspect of clustering. In memory replication, information about the state of user sessions with deployed applications in a GlassFish v2 instance is replicated to a peer instance in the cluster. Each GlassFish v2 instance in a cluster sends session-state information to the next instance in the cluster, its *replication partner*.

As [Figure 7](#) shows, if Instance 1 fails, the load balancer routes the

sessions interacting with Instance 1 to another instance, here Instance 2. If the routed-to instance is the replication partner of the failed GlassFish instance, the sessions are handled there. But if the routed-to instance is not the replication partner, the replication partner transfers the replicated session data to the routed-to instance. In other words, if the load balancer routes the sessions to Instance 3, which is not the replication partner, then Instance 2, which is the replication partner, sends the state data to Instance 3 to handle the sessions.

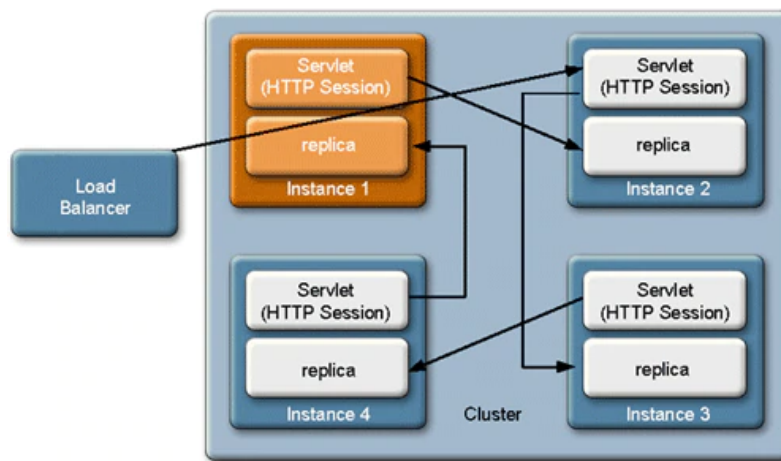
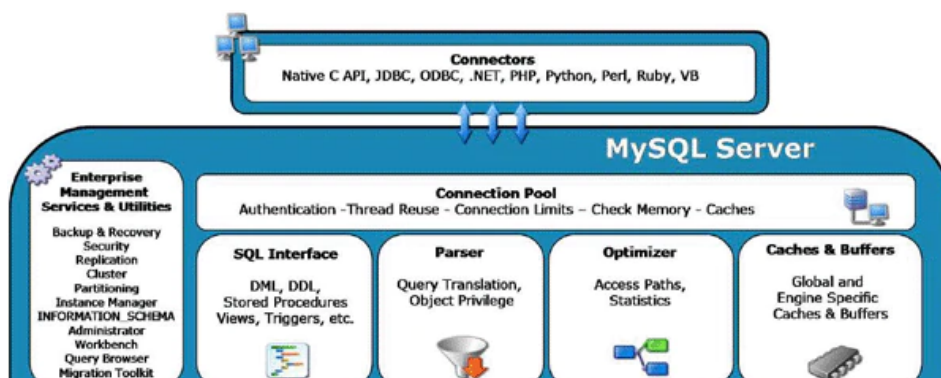


Figure 7. Handling a Failover in a Cluster

You can place cluster instances and their replication partners on different machines to further increase availability. In this way, if one of the machines fails, session data is not lost.

## Flexible Optimization

MySQL includes a pluggable storage engine layer in which it offers many different types of databases, each optimized for a specific type of processing. The storage engine layer is part of the MySQL storage architecture shown in [Figure 8](#).





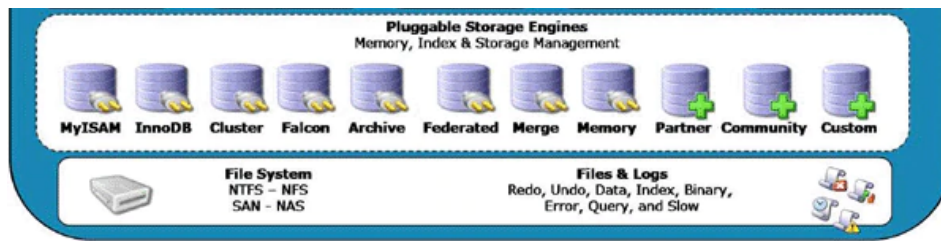


Figure 8. MySQL Storage Architecture

MySQL's pluggable storage engine layer gives you the flexibility to choose the type of database best suited for your needs.

The pluggable storage layer gives you a lot of flexibility in choosing a database best suited for your needs. For example, if you're building a web site that requires high-speed reads and high-speed loads, but doesn't need support for transactions, you can take advantage of a MyISAM database, which is optimized for this type of processing. By comparison, if you're building a site that requires transactions, the InnoDB database is a good choice because of its atomicity, consistency, isolation, and durability (ACID) properties, which guarantee that database transactions are processed reliably. In addition, MySQL offers various types of databases from partners and from the community, each optimized for a particular type of processing.

You can use multiple storage engines in a single application. This is particularly useful in a replication setup where a master copy of a database on one server is used to supply copies, called slaves, to other servers. A storage engine for a table in a slave can be different than a storage engine for a table in the master. In this way, you can take advantage of each engine's abilities.

Only the MySQL database system offers this pluggable storage capability.

### Can Be Used in All Major Environments

MySQL and GlassFish run on all major operating systems and hardware.

MySQL is optimized for Windows, Solaris, and Linux and runs on all major operating systems and hardware. In fact, it's the "M" in the widely used LAMP (Linux, Apache Web Server, MySQL, and PHP)



stack, as well as the SAMP (S=Solaris) and WAMP( W=Windows) stack.

Similarly, GlassFish runs on all major operating systems and hardware, including Windows, Solaris, Linux, MacOS, and AIX.

### Even More Reasons to Use GlassFish

GlassFish v2UR2 offers many other features that make it very attractive for web application development and deployment. These features support the following:

GlassFish offers many other features that make it very attractive for web application development and deployment.

- **Dynamic content.** GlassFish supports dynamic content through Java EE 5 web technologies such as [Java Servlet 2.5](#), [JavaServer Pages \(JSP\) 2.1](#), [JavaServer Pages Standard Tag Library \(JSTL\)](#), and [JavaServer Faces 1.2](#). These technologies provide simple, consistent mechanisms for extending web applications beyond static web pages.
- **Web services.** GlassFish implements the [Metro](#) stack of web services technologies. Metro includes core web services technologies such as [JAX-WS](#) and [JAXB](#) that enable you to develop and use web services. In addition, Metro includes technologies such as [Web Services Interoperability Technologies \(WSIT\)](#) that support secure, transactional web services that can interoperate between Java EE and Windows .NET environments. GlassFish also supports [JAX-RS](#), the Java API for RESTful Web services, and its reference implementation, [Jersey](#). GlassFish supports web services that conform to the SOAP protocol as well as web services that conform to the Representational State Transfer (REST) architectural style.
- **Web 2.0.** GlassFish supports the type of web applications and mashups that are often associated with the term *Web 2.0*. It does this through support for JavaScript and Ajax. GlassFish is compatible with jMaki, a framework for creating JavaScript-centric, Ajax-enabled web applications using Java technology, [PHP](#), or Phobos. Phobos is a lightweight, scripting-friendly, web application

environment that runs on the Java platform. GlassFish also supports "Reverse Ajax/Ajax Push," methodology, which pushes data from the server to the client. This approach, also known as Comet, requires long-lived HTTP connections. The Grizzly framework in GlassFish is used to create and maintain these long-lived HTTP connections.

- **Data persistence.** Data persistence deals with the way that relational data is mapped to Java objects ("persistent entities"), the way that these objects are stored in a relational database so that they can be accessed at a later time, and the continued existence of an entity's state even after the application that uses it ends. GlassFish supports data persistence through a Java Persistence API engine based on Toplink Essentials. The [Java Persistence API](#) provides a simple programming model for entity persistence and includes an API for plugging in different persistence providers.
- **Other languages.** In addition to Java-based web applications, web applications that are developed in languages such as JavaScript, Ruby, and PHP can be deployed on GlassFish.

### **Comparing GlassFish With Tomcat**

GlassFish can provide a better open-source solution for web applications than Tomcat.

Apache Tomcat is an open-source web container that runs Java servlets and JSP pages in web applications. Tomcat is popular in the open source community, and is often used with MySQL. However, it's important to understand that GlassFish can provide an even better open-source solution for web applications than Tomcat, and for many developers might be a better choice to use with MySQL. This section highlights some of the reasons why.

### **More Extensive Web Technology Implementation**

GlassFish is a full-fledged application server that implements all Java EE 5 technologies as well as Web 2.0 technologies.

GlassFish provides a more extensive web technology

implementation than Tomcat. Although Tomcat supports servlet and JavaServer Faces technology, GlassFish goes beyond that, supporting all Java EE 5 technologies, including the following:

- [Web tier technologies](#)
- [Enterprise technologies](#)
- [Web services technologies](#)
- [Management and security technologies](#)

In addition, GlassFish supports Web 2.0 technologies and methodologies such as jMaki and Comet.

Because Tomcat is limited essentially to servlet and JSP technology, you have to install additional software if you want other web tier functionality, or if you want technology that is beyond the scope of the web tier. GlassFish is a full-fledged application server that implements all Java EE 5 technologies as well as Web 2.0 technologies. You do not need to install any additional software to take advantage of technologies and frameworks such as the following:

- [JavaServer Faces](#) technology to create user interfaces for dynamic web applications
- The [Metro](#) technology stack to develop and use web services
- [EJB](#) technology to implement the business logic for enterprise applications
- The [Java Persistence API](#) to enable data persistence for your web applications
- The jMaki framework to quickly build Ajax-enabled web applications

In addition, GlassFish offers Java EE 5-based ease-of-use features such as dependency injection that are not available in Tomcat.

### **Better Administration Tools**

Although Tomcat comes with a number of administration and monitoring tools, such as the Tomcat Web Application Manager, GlassFish provides more comprehensive and easier-to-use administration and monitoring tools. Users can perform almost any

GlassFish-related administration task either from an intelligent CLI or from a web-based Admin Console. The [Admin Console Common Tasks page](#) presents buttons to initiate many of the more commonly performed tasks such as deploying a web application, so that these actions can be performed from a central point.

GlassFish provides more comprehensive and easier-to-use administration and monitoring tools than Tomcat.

Significantly, the Admin Console is enabled for [Java Management Extensions \(JMX\) technology](#) through registered Management Beans (MBeans). This allows you to connect the Admin Console to other JMX tools such as [Java Monitoring and Management Console \(JConsole\)](#) and [VisualVM](#) for more extensive monitoring. In addition, you can extend GlassFish's administration and monitoring features by creating and registering your own custom MBeans.

GlassFish monitoring facilities make it particularly easy to monitor resources such as connection pools. GlassFish also includes a call flow feature that makes it easy to track a transaction as it moves through various GlassFish application server containers.

### **Better Performance**

GlassFish has better HTTP-related performance than Tomcat's HTTP handler.

GlassFish, through its [Grizzly](#) framework, has [better HTTP-related performance than Tomcat's Java-based HTTP connector, Coyote](#), and scales better to handle more requests than Tomcat. Because Tomcat is a web container and GlassFish is a full-fledged application server, their overall performance can't be directly compared. However, GlassFish is an extremely fast application server, as demonstrated by benchmarks such as [SPECjAppServer](#), [1](#).

### **More Flexible Load Balancing**

GlassFish clustering gives administrators more flexibility in terms of

load balancing than Tomcat.

GlassFish, through its clustering capabilities, gives administrators more flexibility in terms of load balancing than Tomcat. In addition, clustering enables a high degree of reliability. If a GlassFish v2 instance in a cluster fails, the session interacting with that instance is rerouted to another instance in the cluster.

### Easy Migration Path

An application that runs in Tomcat will run in GlassFish without modification.

Last but not least, there is an easy migration path from Tomcat to GlassFish. An application that runs in Tomcat will run in GlassFish without modification.

### Summary

Easy to use, easy to administer, integrated with popular development tools, fast, scalable, reliable -- these are all characteristics that make MySQL the world's most popular open-source database. However, the very same characteristics also apply to GlassFish -- an open-source, enterprise-quality, Java EE 5-compliant application server. Those characteristics as well as GlassFish's comprehensive feature set make it an ideal choice to use with MySQL for developing and deploying web applications.

In the [next article in this series](#), you'll see how to take advantage of MySQL and GlassFish features to easily build a CRUD web application that accesses persistent data.

<sup>1</sup>SPEC and SPECjAppServer are registered trademarks of Standard Performance Evaluation Corporation. Results from [www.spec.org](http://www.spec.org) as of 11/03/2008. 2xSun Fire X4150 (8 cores, 2chips) and 1xSun Fire X4150 (4 cores, 1 chip) 1197.10 SPECjAppServer2004 JOPS@Standard; Best result with 8 cores in application-tier of the benchmark: 1xHP BL460c (8 cores,2chips) and 1xHP BL480c (8 cores,2 chips) 2056.27 SPECjAppServer2004 JOPS@Standard; Best result with 2 systems in application-tier of

the benchmark: 2xDell PowerEdge 2950 (8 cores, 2 chips) and  
1xDell PowerEdge R900 (4 chips, 24 cores) 4,794.33  
SPECjAppServer2004 JOPS@Standard.

### For More Information

- [GlassFish and MySQL, Part 2: Building a CRUD Web Application With Data Persistence](#)
- [GlassFish and MySQL, Part 3: Creating and Using a Web Service](#)
- [GlassFish and MySQL, Part 4: Creating a RESTful Web Service and JavaFX Client](#)
- [Java EE Technologies](#)

### Enterprise Versions Are Available Too

In addition to their community-based versions, enterprise versions of MySQL and GlassFish are also available. These versions provide advanced features for enterprises and offer subscription-based services at attractive prices. MySQL offers an all-in-one subscription that includes [MySQL Enterprise Server](#), the enterprise-class version of the MySQL Community Server database. MySQL Enterprise Server provides all of the features in the open-source Community Server and adds monitoring capabilities for maximum uptime and performance, as well as advanced tools for backup, query analysis, and troubleshooting. The subscription offers services such as around-the-clock support with consultation, priority escalation, and access to online knowledge databases. Monthly service updates, quarterly service packs that roll up the monthly service updates, and hot fixes are also available.

The enterprise version of GlassFish is called [Sun GlassFish Enterprise Server](#). It offers subscription-based support service and indemnification as well as features such as High-Availability Database (HADB) technology that ensures 99.999 percent ("five-nines") availability and scalability. Sun GlassFish Enterprise Server and MySQL Enterprise Server are available in a downloadable package called [GlassFish & MySQL Unlimited](#).